تمرین اول

درخت تصمیم

استاد درس: دکتر آرش عبدی هجران دوست

تدریس یار: سید فرید موسوی

نکات تمرین:

- مهلت تحویل ساعت ۲۳:۵۵ ، ۱۴۰۲/۱۲/۲۹
- مهلت ارسال به هیچ وجه قابل تغییر نیست.
- مواردي که بعد از تاریخ فوق ارسال شوند قابل قبول نبوده و نمره اي نخواهند داشت .
- انجام تمرین تک نفره است. لطفا به تنهایی انجام شود، در غیر اینصورت نمره منفی در نظر گرفته خواهد شد.
- کل محتواي ارسالی زیپ شود و نام فایل زیپ ارسالی HW1_StudentNumber_YourName باشد.
- محتواي ارسالی داراي راهنما (read me) جهت تسهیل اجرا باشد.
- زبان برنامه نویسی دلخواه است. (پیشنهاد : پایتون)
- در صورت استفاده از زبان پایتون فایل کد ترجیحا به فرمت ipynb. بوده و فایل کد حتما به صورت اجرا شده آپلود گردد و از وجود خروجی سلول ها اطمینان حاصل نمایید.
- موارد ارسال شده در تاریخی که بعدا مشخص خواهد شد و متعاقبا اعلام میگردد به صورت آنلاین نیز تحویل گرفته خواهند شد.
- صرفا آنچه در LMS طبق تاریخ فوق تحویل داده شده است بعدا به صورت حضوري تست شده و توضیح داده میشود.
- تنها تکالیفی که به LMS و قبل از مهلت ارسال، فرستاده می شوند بررسی خواهند شد.
- در صورت داشتن هر گونه سوال میتوانید سوال خود را در گروه تلگرامی درس مطرح کنید .
- حداقل **یک ساعت قبل از مهلت ارسال** را احتیاطا هدف قرار دهید، تا مشکلات غیرقابل پیش بینی مانند موارد زیر باعث عدم آپلود پاسخ ها در LMS و ارسال آنها از طریق ایمیل نشوند : (قطعی اینترنت، تنظیم نبودن دقیق ساعت سایت با ساعت گرینویچ، کرش سیستم عامل و نیاز به فرمت، بارش زیباي شهاب سنگ از آسمان و ...)

لطفاً از چت جی پی تی و مدل های هوش مصنوعی مشابه برای پیاده سازی استفاده نکنید، شما خودتان دانشجو هوش مصنوعی هستید 😊

بخش اول (پیاده‌سازي اوليه)

در ابتدا، ميخواهيم رده‌بند(classifier) درخت تصميم را از بيخ و بن (بدون استفاده از کتابخانهي آماده) ، براي داده‌هاي گسسته، مطابق شبه کد ارائه شده در اسلايدهاي کلاس، خودمان پياده‌سازي نماييم. بهتر است که تاکيد کنيم، اين بخش بسيار حائز اهميت ميباشد زيرا در ادامه پروژه با کد پياده سازی شده خودتان قرار است کار کنيد.

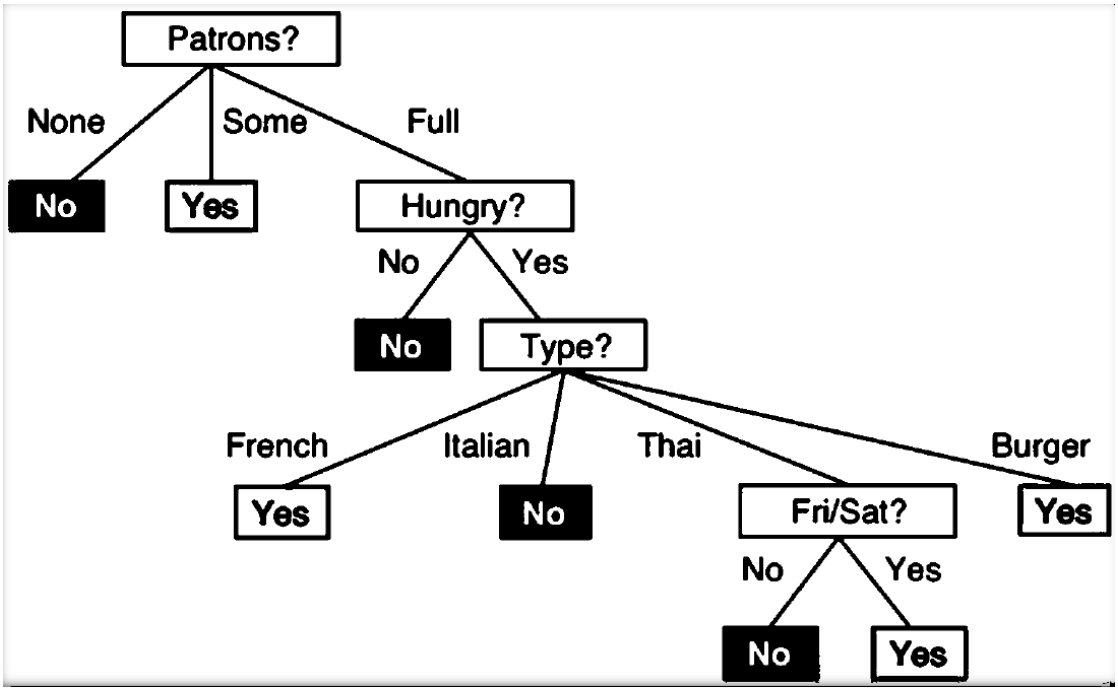
function DECISION-TREE-LEARNING(*examples*, *attributes*, *parent_examples*) **returns** a tree

if *examples* is empty **then return** PLURALITY-VALUE(*parent_examples*)
else if all *examples* have the same classification **then return** the classification
else if *attributes* is empty **then return** PLURALITY-VALUE(*examples*)
else
  $A \leftarrow \text{argmax}_{a \in attributes}$ IMPORTANCE(a, *examples*)
  tree $\leftarrow$ a new decision tree with root test A
  for each value v_k of A **do**
    $exs \leftarrow \{e : e \in examples \text{ and } e.A = v_k\}$
    subtree $\leftarrow$ DECISION-TREE-LEARNING(exs, *attributes* $- A$, *examples*)
    add a branch to *tree* with label $(A = v_k)$ and subtree *subtree*
  return *tree*

براي اطمينان از صحت پياده سازي صورت گرفته، ميتوانيد داده هاي ۱۲ تايی مثال رستوران (مطرح شده در کلاس درس) را مورد بررسی قرار دهيد.

Example	Input Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	$y_1 = Yes$
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	$y_2 = No$
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	$y_3 = Yes$
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10–30*	$y_4 = Yes$
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*>60*	$y_5 = No$
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	$y_6 = Yes$
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	$y_7 = No$
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	$y_8 = Yes$
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*>60*	$y_9 = No$
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	$y_{10} = No$
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	$y_{11} = No$
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	$y_{12} = Yes$

براي اين كار تمام ۱۲ داده را به عنوان مجموعه آموزشی در نظر بگيريد(بدون مجموعه‌ي آزمايشی مجزا) و الگوريتم را براي اين داده‌ها اجرا نماييد و سعی كنيد ساختار درخت آموزش ديده شده را نمايش دهيد، خروجی صحيح مطابق تصوير زير ميباشد(البته ممكن است دو ويژگی آنتروپی يكسانی داشته باشند و ساختار درخت شما در يک لايه متفاوت باشد.)



حتما نياز نيست درخت به صورت گرافيكی و شماتيک باشد ميتوانيد از كتابخانه‌های آماده و يا ساده ترين حالت print درخت استفاده كنيد.

بخش دوم (دسته‌بندي)

در اين مرحله ميخواهيم با استفاده از رده‌بند درخت تصميمی كه در مرحله قبل پياده سازي كرده‌ايد، يک مسئله دسته‌بندي با داده‌هاي واقعی را حل كنيم. براي اين كار از مجموعه دادگان اين لينک استفاده ميكنيم كه در كنار اين فايل با نام Dry_Bean_Dataset.csv نيز آمده است كه يک مجموعه داده براي تشخيص دانه های خشک(حبوبات) با استفاده از تعدادي ويژگی ميباشد.**قابل ذكر هست كه مسئله چندكلاسه ميباشد.**

در ابتداي كار ميبايست داده‌ها را به دو بخش داده‌هاي آموزشی(train set) و داده‌هاي آزمايشی(test set) تقسيم كنيد. نحوه بخش بندي دادهها به دو بخش آموزش و آزمايش به صورت كاملا اختياري و به دلخواه خودتان است (مثلا ۹۰-۱۰ ، ۲۰-۸۰ و ...)

همانطور كه ميدانيد ورودي‌هاي درخت تصميم بايد به صورت گسسته باشد. براي گسسته سازي ورودي‌هاي از نوع پيوسته (مانند مجموعه داده فعلی) روشهاي مختلقی وجود دارد. ساده‌ترين ايده آن است كه براي چنين ويژگی هايی، بازه مينيمم تا ماكزيمم اعداد در مجموعه آموزشی را به تعدادي بازه مساوي تقسيم كنيد (چه تعداد؟ تعدادهاي مختلف را

آزمايش كنيد) ايده هاي بهتر براي گسسته سازي مانند مرتب سازي و انتخاب نقاط برش در هر گره از درخت بر اساس نمونه هایی که در آن گره حاضرند را نیز امتحان کنید. همچنين ميتوانيد ايده هاي مطرح شده در کلاس یا ایده‌هاي جديد و خلاقانه خود نيز استفاده كنيد و نتايج آن را با حالت هاي قبل (بازه هاي مساوي يا انتخاب نقاط برش بر حسب مرتب سازي) مقايسه كنيد.

سپس با استفاده از الگوريتم نوشته شده درخت تصميم در بخش قبلی آموزش مدل را بر روي دادههاي آموزشی انجام دهيد(در نظر داشته باشيد براي پياده سازي درخت تصميم **نبايد** از توابع آماده استفاده كنيد، لذا فرمول آنتروپی و ... را بايد خودتان پياده كنيد. استفاده از توابع آماده براي قسمتهاي بعدي بلامانع است (و حتی توصيه ميشود) مثلا براي خواندن اكسل، نمايش گرافيکی خروجی درخت براي درك شهودي بهتر از فرآيند نحوه تصميم گيري درخت تصميم (كه الزامی نيست)، نمايش دقت خروجی و ... .

همانطور که در کلاس گفته شد معیار هایی برای انتخاب بهترین ویژگی ها وجود دارد، شما با معیار آنتروپی و information gain آشنایی دارید، از آن ها برای feature selection استفاده کنید و نتایج را بررسی کنید، معیار های دیگری نیز در این تسک موجود میباشد برای مثال از معروفترین آن ها میتوان به gini index اشاره کرد. برای بهتر شدن نتیجه خود میتوانید از این معیار استفاده کنید و نتیجه را با دیگر معیار های موجود مقایسه کنید. اگر نتایج متفاوت بود، دلیلی برای این موضوع میتوانید بیان کنید.

براي جلوگيري از بيش‌برازش(overfitting) از روشهاي توقف زودهنگام(early stopping, pre-pruning) براي جلوگيري از برگ‌هاي با تعداد نمونه‌هاي بسيار كم (يا خيلی gainكوچک) قبل از ساخت كامل درخت و روش‌هاي هرس كردن(pruning, post-pruning) پس از آموزش كامل و ساخت درخت، مورد استفاده قرار ميگيرد. يک روش براي توقف زودهنگام و يک روش براي هرس كردن را پياده سازي كنيد و با مقادير مختلف آزمايش كنيد. نتايج اين دو روش را با يكديگر مقايسه كنيد. دلیلی برای تفاوت این دو پیدا میکنید؟

بعد از فرآيند آموزش درخت تصميم، مقادير زير را براي داده‌هاي آموزشی و داده‌هاي آزمايشی همه‌ي حالت‌هاي گفته شده محاسبه كنيد:

۱) صحت(accuracy) ۲) دقت(precision) ۳) فراخوانی(recall) ۴) معیار f1

۵) ماتریس درهم ریختگی(confusion matrix) ۶) نمودار ROC

قابل ذکر هست مقدار شماره ۶ اختیاری میباشد.

دقت كنيد كه در آزمايشها يک مقدار حداقلی از مقادير دقت مد نظر است (ميتوانيد درخت تصميم نوشته شده با كتابخانه آماده را به عنوان اين مقدار مناسب در نظر بگيريد) چنانچه درخت شما مقادير دقت قابل قبولی نداشت (در مقايسه با درخت آماده موجود در كتابخانه خيلی تفاوت زيادبود) حتما سعی كنيد دقت را افزايش و نتايج را بهبود دهيد. (با تغيير مجموعه هاپير پارامترها چه در توزيع داده ها و چه در خود مدل و ... .) در غير اين صورت منجر به كسر نمره خواهد شد.

بخش سوم (تقریب تابع)

در این بخش میخواهیم با استفاده از درخت تصمیم تقریب تابع (مسئله‌ي رگرسیون) انجام دهیم. براي این کار از مجموعه داده‌ي housing.csv موجود در کنار این فایل، استفاده نماییم که یک مجموعه داده براي تخمین میانگین قیمت یک خانه در کالیفرنیا با استفاده از تعدادي از ویژگی‌هاي آن خانه میباشد. براي اطلاعات بیشتر درباره این مجموعه داده به این لینک مراجعه نمایید.

در ابتداي کار نیاز است پس از تقسیم این مجموعه داده به مجموعه آموزشی و آزمایشی، یک پیش پردازش مناسب بر روي آن انجام داده(دقت کنید که مجموعه داده دارای missing value و ... میباشد) و با ویژگی‌هاي آن بیشتر آشنا شوید و براي ویژگی‌هاي پیوسته، گسسته سازي به مانند بخش قبلی انجام دهید.

در گام بعدي نیاز است تا درخت نوشته شده در بخش قبلی را با کمی تغییرات (که در کلاس درس به آن اشاره شده است)، به یک درخت تصمیم مسئله رگرسیون تغییر دهید.

در برگ‌ها نیز از روش میانگین‌گیري، میانه‌گیري و fit کردن تابع استفاده نمایید. (نتایج این سه روش را می خواهیم در ادامه با یکدیگر مقایسه نماییم)

میتوان مجددا از هرس کردن براي جلوگیري از بیش برازش (در صورت لزوم) استفاده کنید.

براي ارزیابی این مدل، معیارهاي ارزیابی براي مسئله رگرسیون را مورد مطالعه قرار دهید و هم براي دادههاي آموزشی و هم براي داده‌هاي آزمایشی این معیارها را محاسبه و تحلیل نمایید.

<u>آنچه تحویل داده میشود:</u>

۱) کداجرایی برنامه با توضیحات لازم براي اجرا

۲) درختی که براي مرحله دوم و سوم پیدا کرده اید (میتوانید گرافیکی نمایش دهید (به هر نحوي که میتوانید) یا به صورت Text با پروتکلی که توضیح میدهید و قابل فهم باشد، بتوان فهمید در هر گره کدام ویژگی با چه مقادیري خروجی تست شده اند و زیر شاخه هایش کدامند.

۳) گزارش کاملی از مسیر انجام کار، چالش‌هایی که مواجه شده‌اید، اجراهایی که گرفتید و نتایجی که حاصل شده است. گزارش کار از اهمیت بالایی برخوردار است، حجم آن و فرمت استاندارد آن اهمیت ندارد، اما باید نشان دهنده مسیر انجام پروژه، چالش‌ها، راه حل‌ها و نتایج کار شما باشد.